

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. - 18. (Cancelled):

19. (New): A method of illustrating a software process in a diagram, the software process including software objects and operations, each operation being associated with a respective one of the software objects, and control flow mechanisms determining a control flow of the software process, each operation being associated with a respective one of the control flow mechanisms, the diagram being characterized by a horizontal direction and a vertical direction, the method comprising:

placing in the diagram, for each control flow mechanism, a control flow segment representing the control flow mechanism, and further placing the control flow segments in the diagram in series so as to form a timeline extending longitudinally in the horizontal direction, the timeline illustrating the control flow of the software process including a sequence of the operations, each control flow segment defining at least one respective vertical column;

placing in the diagram, for each software object, an elongated shape representing the software object, the elongated shape extending longitudinally in the horizontal direction, the elongated shape containing text specifying the software object, and further placing each elongated shape in the diagram so as to be parallel to the

timeline and spaced apart in the vertical direction from the timeline;

placing in the diagram, for each operation, a compact shape representing the operation, the compact shape identifying an operation type of the operation, the compact shape containing a symbol particularly specifying the operation, and further placing each compact shape in the diagram so as to be adjacent the elongated shape representing the software object associated with that operation, thereby illustrating the association of that operation with that software object, and further placing the compact shape in the vertical column defined by the control flow segment representing the control flow mechanism associated with that operation, thereby also illustrating the association of that control flow mechanism with that operation;

whereby the software process is illustrated in the diagram.

20. (New): The method according to claim 19 wherein a first one of the operations is associated with a first one of the software objects, and a second one of the operations is associated with a second one of the software objects, the first operation producing data based on the first software object, the second operation operating on the second software object based on the data, the method further comprising:

further placing in the diagram first and second ones of the elongated shapes respectively representing the first and second software objects, such that a respective part of each of those elongated shapes is contained in a particular one of the vertical columns, and such that those elongated shapes are spaced apart in the vertical

direction;

further placing in the diagram, in the vertical column containing the respective parts of the first and second elongated shapes, first and second ones of the compact shapes respectively representing the first and second operations, the first compact shape identifying the operation type of the first operation as producing the data from the first software object, the second compact shape identifying the operation type of the second operation as operating on the second software object; and

further placing in the diagram a line connecting the first and second compact shapes thereby illustrating that the operation of the second operation on the second software object is based on the data produced by the first operation based on the first software object.

21. (New): The method according to claim 20 wherein the first and second compact shapes are spaced apart in the horizontal direction thereby illustrating that the second operation follows the first operation in the sequence of the operations.

22. (New): The method according to claim 21 wherein the first compact shape is spaced apart in the vertical direction from the first elongated shape thereby illustrating that the first operation does not modify the first software object, and wherein the second compact shape touches the second elongated shape thereby illustrating that the second operation modifies the second software object.

23. (New): The method according to claim 20 wherein a third one of the operations is associated with a third one of the software objects, the third operation producing data based on the third software object, and wherein the control flow mechanism associated with the third operation is conditional upon the data produced by the third operation, the method further comprising:

further placing in the diagram a line connecting the compact shape representing the third operation and the control flow segment which represents the control flow mechanism associated with the third operation, that compact shape identifying the operation type of the third operation as producing the data from the third software object, thereby illustrating that the control flow mechanism represented by that control flow segment is conditional upon the data produced by the third operation based on the third software object.

24. (New): The method according to claim 23 wherein fourth and fifth ones of the software objects are related, the method further comprising:

further placing in the diagram fourth and fifth ones of the elongated shapes respectively representing the fourth and fifth ones of the software objects, such that a respective part of each of those elongated shapes is contained in a particular one of the vertical columns, and such that those elongated shapes are spaced apart in the vertical direction;

further placing in the diagram a relationship symbol adjacent the fourth

Serial No.: 10/804,033  
Office Action dated June 3, 2008  
Amendment dated October 3, 2008

elongated shape, and further connecting the relationship symbol to the fifth elongated shapes with respective lines, thereby illustrating the relationship of the fourth and fifth software objects.

25. (New): The method according to claim 24 wherein the fifth software object is related to the fourth software object by one of inheritance, data content, encapsulation/breakout, or interface.

26. (New): The method according to claim 19 wherein a particular one of the software objects is associated with a plurality of the operations, that plurality of operations being associated with a respective plurality of the control flow mechanisms, whereby the control flow segments respectively representing such plurality of control flow mechanisms define a respective plurality of vertical columns, and wherein each one of such plurality of vertical columns contains at least one of the compact shapes respectively representing that plurality of operations, and wherein each one of such plurality of columns further contains a respective portion of the elongated shape representing that software object.

27. (New): The method according to claim 24 wherein the software process further includes a list assignment or a parameter specification, the method further comprising

placing in the diagram a further shape containing details of the list assignment or the parameter specification, respectively.

28. (New): The method according to claim 27 wherein the software process further includes a mathematical expression, the method further comprising placing in the diagram an even further shape containing the mathematical expression.

29. (New): The method according to claim 28 wherein at least one of the control flow mechanisms is selected from a group comprising: looping; conditional branching; nested looping; nested branching; exception branching; and thread handling.

30. (New): The method according to claim 29 wherein the text contained by at least one of the elongated shapes specifies that the software object represented by that elongated shape includes: an array; a hash; a database; a table; a file; a queue; a stack; a tree structure; or a software variable.

31. (New): The method according to claim 30 wherein the symbol contained by at least one of the compact shapes specifies the operation represented by that compact shape as including: sorting; selecting; parsing; substitution; formatting; copying; making an assignment; making a state change; making a computation; or returning a value.

32. (New): A graphical user interface of a computer aided design software tool in a computer system for illustrating a software process according to the method of claim 19, the graphical user interface comprising:

- a grid for the placement of the control flow segments, each of the compact shapes, and each of the elongated shapes, in the diagram;

- a first set of activatable controls for placing each of the control flow segments on the grid;

- a second set of activatable controls for placing each of the compact shapes on the grid, the graphical user interface presenting to the user a dialog box for entry of the symbol specifying the operation represented by that compact shape when that control is activated; and

- a third activatable control for placing each of the elongated shapes on the grid, the graphical user interface presenting to the user a dialog box for entry of the text specifying the software object represented by the elongated shape when that control is activated.

33. (New): The graphical user interface according to claim 32 wherein at least one of the control flow mechanisms is selected from a group comprising: looping; conditional branching; nested looping; nested branching; exception branching; and thread handling, and the first set of activatable controls includes a specific activatable control for that control flow mechanism.

34. (New): The graphical user interface according to claim 33 wherein the dialog box for entry of the text specifying the software object is for entry of the text specifying that the software object includes: an array; a hash; a database; a table; a file; a queue; a stack; a tree structure; or a software variable.

35. (New): The graphical user interface according to claim 34 wherein the operation type of at least one of the operations includes: sorting; selecting; parsing; substitution; formatting; copying; making an assignment; making a state change; making a computation; or returning a value, and the second set of activatable controls includes a further specific activatable control for that operation type.

36. (New): A computer program product for use with the computer system, the computer program product comprising a tangible computer-readable medium having encoded thereon computer-readable code for implementing the graphical user interface according to claim 32.

37. (New): The computer program product according to claim 36 wherein the tangible computer-readable medium comprises a diskette, a CD-ROM, a fixed disk, or a memory device including a semiconductor, magnetic, or optical memory device.